Thank you again for speaking with us. As the next step, we’d like you to complete a brief technical assessment that mirrors a problem we solve every day.

Context:
Our platform ingests high-frequency telemetry from IoT devices via a message broker (choose **NATS**, **Kafka**, or **RabbitMQ**).

Each message has the following structure:

```
{
  "imei": "string",
  "latitude": float64,
  "longitude": float64,
  "device_time": int64 // milliseconds
}
```

Your Task (Go):

1. Implement, in Golang, a service that consumes these messages.
2. For each imei, compute the cumulative distance travelled using the latitude/longitude stream and store the result (in Redis or a database of your choice).

Constraints & Expectations:

- High throughput: multiple instances of your service must be able to run concurrently.
- Ordering: messages from each device arrive in order, but when a device gets disconnected and later reconnects, it sends a batch of events (still in order). Your service will receive them one by one from the message broker.
- We receive the events from all devices through the same topic/subject/key in the message broker.

Please include a concise **README** explaining:

- Your solution and the bottlenecks you addressed
- How to build and run the application locally

When you’re finished, please send us a link to a public repository (or a zip file) containing your code and README.
Feel free to reach out if you have any questions. We’re looking forward to seeing your solution.